LU16.A02 – BDD in der eigenen Sprache

In dieser Aufgabe wendest du das Prinzip von **Behaviour Driven Development (BDD)** selbständig in deiner bevorzugten Programmiersprache an. Du wählst ein geeignetes Framework, setzt ein kleines Beispielprojekt um und reflektierst deine Erfahrungen in einem kurzen Dokument.

Ziel

- Du verstehst, wie BDD in verschiedenen Programmiersprachen umgesetzt wird.
- Du kannst eigenständig ein BDD-Framework auswählen, installieren und anwenden.
- Du reflektierst, welche Vor- und Nachteile BDD in deinem persönlichen Arbeitskontext hat.

Auftrag 1 – Recherche und Setup

1. **Wähle eine Programmiersprache oder ein Framework**, das du bereits gut kennst (z. B. Java, JavaScript, C#, PHP, Ruby, Go, Swift …). 2. **Recherchiere**, welches BDD-Framework dort üblich ist. Beispiele:

- Python → *behave* oder *pytest-bdd*
- Java → *Cucumber* oder *JBehave*
- JavaScript / TypeScript → *Cucumber.js* oder *Jest Cucumber*
- C# → *SpecFlow*
- Ruby → *RSpec* oder *Cucumber*
- Go → *Godog*

3. **Installiere das Framework** und richte eine einfache Projektstruktur ein (ähnlich wie bei Behave mit ``features/`` und ``steps/``). 4. **Dokumentiere kurz**, welches Framework du gewählt hast und wie du es installiert hast (z. B. welche Befehle oder Tools du verwendet hast).

Auftrag 2 – Eigenes BDD-Miniprojekt

Erstelle ein eigenes kleines Beispielprojekt (max. 30–40 Zeilen Code insgesamt):

Beispielthemen:

- Taschenrechner mit Addition und Subtraktion
- Login mit Benutzername/Passwort

- Einkaufskorb mit einem Produkt
- Temperatur- oder Währungsumrechnung
- API-Endpoint-Test (wenn du Erfahrung mit HTTP-Requests hast)

Vorgehen:

```
1. Formuliere **mindestens ein Feature** und **zwei Szenarien** in Gherkin-Syntax (''Given / When / Then'').
2. Implementiere die zugehörigen Schrittdefinitionen im Code.
3. Implementiere den minimalen Produktivcode, der die Tests erfüllt.
4. Führe die Tests aus und dokumentiere das Resultat (Screenshot oder Test-Log).
```

Auftrag 3 – Reflexion

Erstelle ein **Reflexionsdokument (ca. ½ Seite)** mit folgenden Fragen:

Teil A – Verständnis und Umsetzung

```
1. Wie hast du dein Framework ausgewählt und warum?
2. Wie aufwändig war die Einrichtung?
3. Wie unterscheidet sich der Aufbau deines Frameworks von ''behave'' (z. B. Syntax, Ordnerstruktur, Ausführung)?
4. Welche Stolpersteine gab es beim ersten Setup oder beim Schreiben der Tests?
```

Teil B – Einschätzung und Nutzen

```
5. Welche Vorteile siehst du im BDD-Ansatz gegenüber klassischen Unit Tests?
6. Gibt es in deinem Ausbildungsbetrieb oder Projektumfeld realistische Anwendungsmöglichkeiten für BDD?
7. Was würdest du an der Methode oder am Framework verbessern oder vereinfachen?
8. Würdest du BDD im Berufsalltag einsetzen – warum (nicht)?
```

Speichere deine Reflexion als **PDF** mit dem Namen:
LU16_A02_BDD_Reflexion_VornameNachname.pdf

Abgabe in Moodle

- Projektdateien (Feature, Steps, Code) als ZIP oder Repository-Link
- Reflexions-PDF